

WHAT IS CLAIMED IS:

1. A system for graphical display of an object created by an application program running under an operating system, comprising:

a graphics resource component adapted to display the object independently of the operating system;

a proxy component, which associates the object with the graphics resource component and invokes methods of the graphics resource component to display the object; and

a peer component, adapted to receive events pertaining to the object and route the events to the proxy component.

2. The system as recited in claim 1, wherein the peer component is independent of the operating system, and emulates the behavior of a second peer component that employs the windowing system of the operating system.

3. The system as recited in claim 1, wherein the object is part of a graphical user interface (GUI) associated with the application program.

4. The system as recited in claim 3, wherein a look and feel of the GUI is independent of the operating system.

5. The system as recited in claim 1, wherein the application program is written in Java programming language.

6. The system as recited in claim 1, wherein the proxy extends an existing class of software components belonging to the Swing application program interface (API).

7. The system as recited in claim 1, wherein the object is part of a layout, and the association of the object with the graphics resource component establishes a parent-child relationship between the layout and the graphics resource component.

8. The system as recited in claim 7, wherein the parent-child relationship between the layout containing the object and the graphics resource component allows the graphics resource component to draw over an existing image of the object drawn with the aid of the windowing system of the operating system.

9. A method for graphical display of an object created by an application program running under an operating system, comprising:

utilizing a graphics resource component adapted to display the object independently of the windowing system of the operating system;

creating a proxy component and establishing an association between the object and the graphics resource component via the proxy component;

receiving events pertaining to the object in a peer component and routing them to the proxy component; and

in response to the events, invoking methods of the graphics resource component via the proxy component to display the object.

10. The method as recited in claim 9, wherein the peer component is independent of the operating system, and the method further comprises emulating the behavior of a second peer component that employs the windowing system of the operating system.

11. The method as recited in claim 9, wherein the object is part of a graphical user interface (GUI) associated with the application program.
12. The method as recited in claim 11, wherein a look and feel of the GUI is independent of the operating system.
13. The method as recited in claim 9, wherein the application program is written in Java programming language.
14. The method as recited in claim 9, wherein the proxy extends an existing class of software components belonging to the Swing API.
15. The method as recited in claim 9, wherein the object is part of a layout, and the method further comprises using the association of the object with the graphics resource component to establish a parent-child relationship between the layout and the graphics resource component.
16. The method as recited in claim 15, further comprising using the parent-child relationship between the layout containing the object and the graphics resource component to allow the graphics resource component to draw over an existing image of the object drawn with the aid of the windowing system of the operating system.
17. A computer-readable storage device, comprising:
 - a windows-based operating system;
 - an application program running under the operating system;

a graphics resource component adapted to display an object created by the application program independently of the windowing system of the operating system, by:

creating a proxy component and establishing an association between the object and the graphics resource component via the proxy component;

receiving events pertaining to the object in a peer component and routing them to the proxy component; and

in response to the events, invoking methods of the graphics resource component via the proxy component to display the object.